

Mano a Mano:

A primer on Hand to hand combat in action videogames

Luis Barriga, Rob Gallerani

Before we start

- ◆ Please turn off cell-phones and pagers.
- ◆ Save your questions for Q&A after lecture.
- ◆ Turn in feedback forms before you leave.

Introduction

◆ Purpose of Lecture

- ◆ Offer a survey of videogame hand-to-hand combat systems and their evolution through time
- ◆ Provide design theory for combat systems centered around the user experience
- ◆ Illustrate the close link between animation and design in combat systems

Lecture at a glance

- ◆ Hand-to-hand combat throughout videogame history
- ◆ Elements of combat design
- ◆ Animation considerations

Hand-to-hand Combat Throughout Videogame History

- ◆ Side-scrolling action games
- ◆ The brawler revolution
- ◆ Brawlers go 3D
- ◆ The bleeding edge
- ◆ Not focusing on pure 1 on 1 fighters

Early Side-scrolling Action Games

- ◆ Highly abstracted combat
- ◆ Touch–damage model
 - ◆ The player's character is always vulnerable unless attacking
 - ◆ The enemy is always attacking
- ◆ One-hit kills not uncommon
- ◆ Heavy timing element

Early Side-scrolling Action Games



The Brawler Revolution

- ◆ **Brawlers made combat representative**
 - ◆ An opponent is hit, he reels back
 - ◆ An opponent is hit hard, he falls to the floor
 - ◆ Symmetry between player's character and enemies
 - ◆ 2.5D – limited z-axis movement
- ◆ **Punch/Kick controls**
 - ◆ Satisfied the obvious combat-hero fantasy
 - ◆ Mostly cosmetic or minor in distinction

The Brawler Revolution

- ◆ **Brawlers popularized combos**
 - ◆ Combos consist of the stringing together of moves to form chains of attacks
 - ◆ They are as much of an innovation as they are a necessity in the brawler model. Without hit-hit-hit-knockdown, the system breaks down into hit-hit-hit-hit, etc.
 - ◆ Combos provided games with a cadence similar to an action-movie brawl

The Brawler Revolution

- ◆ Moderate design evolution
 - ◆ Most brawlers did not innovate greatly
 - ◆ Small incremental innovations were common but design risks were few even towards the end of the cycle of 2D machines

The Brawler Revolution

- ◆ **Double Dragon: the notable exception**
 - ◆ Most DD games brought something new to the table
 - ◆ DDI wrote all the rules
 - ◆ DDII experimented with facing-relative controls (Billy and Jimmy always punched forwards and kicked backwards and the buttons switched with the player's facing)
 - ◆ DDIV added blocking and countering among other things

The Brawler revolution



Brawlers go 3D

- ◆ Not an early adopter's genre
 - ◆ Brawlers stayed 2D for a long time
 - ◆ Rendering the quantity of characters involved represented a technical challenge
 - ◆ Technology leap was more than just graphics, 3D movement was huge jump from traditional 2.5 D games

Brawlers go 3D

- ◆ Design evolved as well
 - ◆ From Punch/Kick to Fast/Hard
 - ◆ Distinct combo effects
 - ◆ Group hits
 - ◆ Stuns
 - ◆ Juggles
 - ◆ Design evolution was partly prompted by tech advancement (360 degree movement required group hits, true vertical movement suggests juggles,etc.)

The Bleeding Edge

- ◆ Surprisingly experimental
 - ◆ For an area of design that's been so widely explored, recent or upcoming games have shown a great deal of design innovation
 - ◆ Mark of Kri (targetting system)
 - ◆ Devil May Cry (timing element)
 - ◆ Dynasty Warriors (mass combat)

Elements of Combat Design

- ◆ Combat is often described in visceral terms
 - ◆ Tight
 - ◆ Button-mashy
 - ◆ Sloppy
 - ◆ Unforgiving
 - ◆ Gratifying

Elements of Combat Design

- ◆ These terms can describe the user experience accurately
- ◆ They don't tell us much about how to implement viable combat systems

Elements of Combat Design

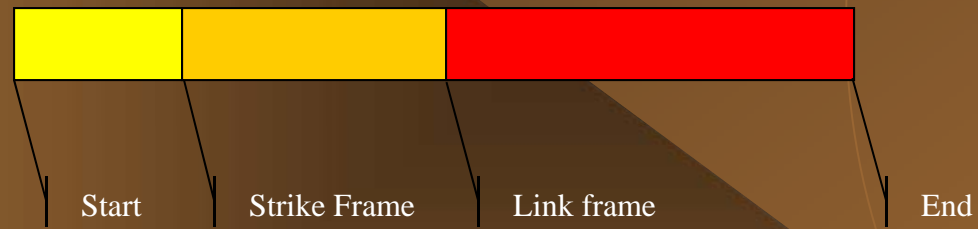
- ◆ Responsiveness
- ◆ Depth
- ◆ Feedback
- ◆ Cadence

Responsiveness

- ◆ 1. User input gets processed in a timely manner
- ◆ 2. Valid user input does not get thrown out

Responsiveness

Attack animation

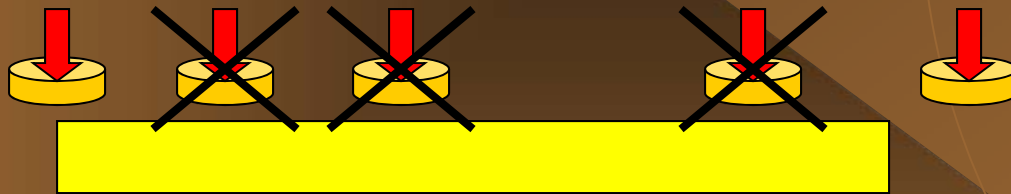


Responsiveness



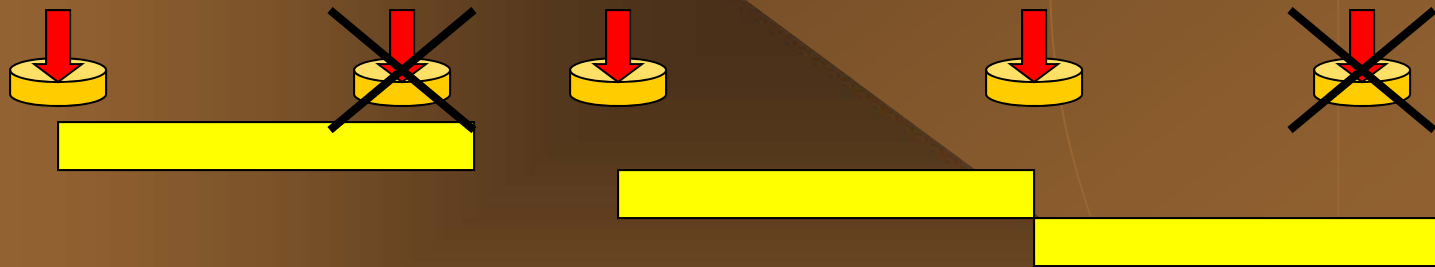
Responsiveness

- ◆ User input is expected after a process is finished
- ◆ What is the process in our combat system? The attack's animation?



Responsiveness

- ◆ What do combos look like in a system where we only accept input after each animation is finished?



Responsiveness

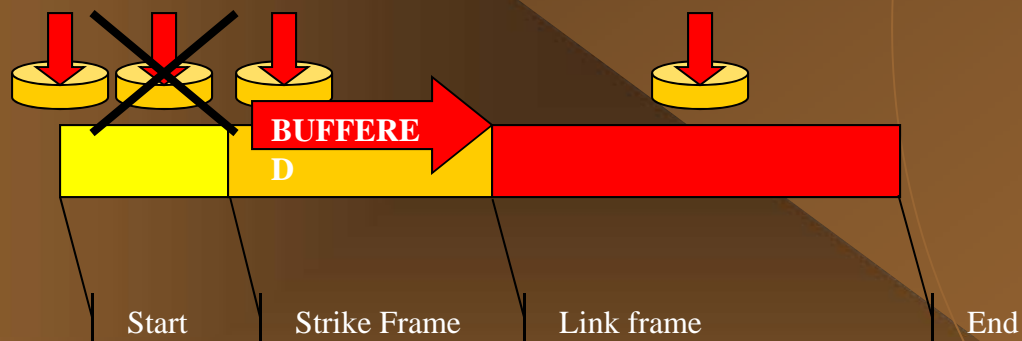


Responsiveness

- ◆ How do we fix this?
 - ◆ Temptation is to simply speed up the animations
 - ◆ Don't do this!

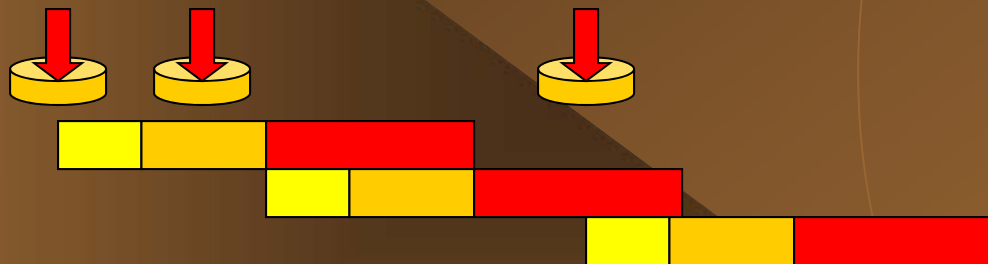
Responsiveness

- ◆ From player's POV, process = attack (players don't care if your animation is done)



Responsiveness

- ◆ What do combos look like in our new system?

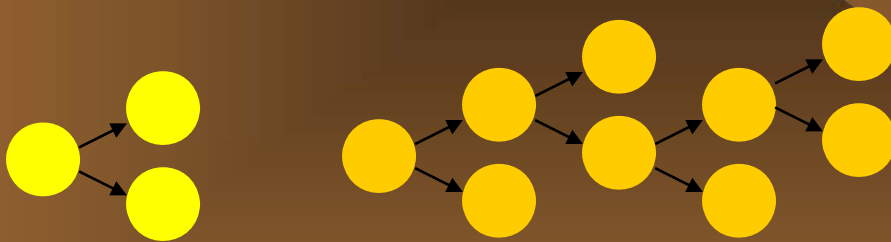


Responsiveness



Depth

- ◆ Range of choices
 - ◆ Combat is interesting only if the user is making interesting choices
 - ◆ All combat systems have the minimal decision tree of attack/don't attack. Systems with distinct combos have deeper trees.



Depth

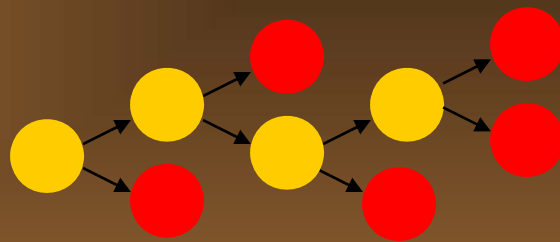
- ◆ **Animation note:**

- ◆ Any moment where a decision might be made and multiple animations can be performed require neutral animations with proper facing and leading (always keep in mind where the player can go from that state).

Depth

◆ Combo trees

- ◆ Distinct hit-types/effects can be applied to different nodes to create distinct purposes for each combo.
- ◆ If non-leaf nodes are all equivalent, then each combo can be evaluated by the user in terms of its length and the effect of the leaf node



Depth

◆ Combat hit-types

- ◆ Normal
- ◆ Knockdown
- ◆ Stun
- ◆ Juggle
- ◆ Group hit
- ◆ Knock-back

Depth



Depth

◆ Matching hit-types to combos

- ◆ When is the user going to perform this action?
- ◆ E.g. a group-hit will usually be performed when surrounded, therefore it should be matched to a short string
- ◆ Hit-types that open up more combo opportunities (stun or juggle) are often used in order to get maximum combo length and long strings match them well

Depth

◆ Getting muscle memory involved

- ◆ Associating a series of repetitive movements as one action gets muscle memory involved in a way similar to typing or playing music
- ◆ This allows for complex and deep controls without involving a large amount of buttons
- ◆ This also gives combat a natural learning curve as, given enough time, simple combinations will be learned by repetition (gesture-based systems work the same way)

Depth

- ◆ **Non-combo based depth**
 - ◆ The following also enhance the system's decision space in addition to using combo-trees
 - ◆ Blocks
 - ◆ Counters
 - ◆ Throws
 - ◆ Resource management (limited-use weapons, special bars, etc.)

Feedback

- ◆ The user is aware of
 - ◆ Whether an action succeeds or not
 - ◆ When said action takes effect
 - ◆ What the consequences of that action are
- ◆ The above can be communicated in different ways
 - ◆ Implied/unobtrusively
 - ◆ Clearly and concisely
 - ◆ Over the top, gratifying

Feedback

◆ Visual

- ◆ Hit animations
- ◆ Knockdown animations
- ◆ Hit indicators
- ◆ HUD updates
- ◆ Text

◆ Aural

- ◆ SFX

Feedback

◆ Hit animations

- ◆ A character gets hit, they play the appropriate animation
- ◆ Effective hit animations reel characters back very quickly
- ◆ Different hit-types need different animations
- ◆ Conventionally characters cannot perform actions while playing hit animations
- ◆ Can be location/direction sensitive (high, medium, low, front/back)

Feedback

◆ Hit animations: Timing

- ◆ If hit animations are shorter in duration than a standard attack animation, it will be possible for quick moves to interrupt combos.
- ◆ If hit animations are too long it might be possible for the player to stun lock the opponent by never delivering a finishing move, just restarting a combo over and over.
- ◆ It is important to have a shared metric for standard attacks and hit animations to prevent these issues from occurring.

Feedback

◆ Timing (continued)

- ◆ If blocking is synched with attacks, a shared metric with blocking is needed as well.
- ◆ Even though most moves will tend to be quick and concise, combo finishers provide a great opportunity for adding flourish. Combo finishers usually disable the opponent for a decent duration, allowing for more involved animations to be used.

Feedback

◆ Hit indicators

- ◆ Sparks, stars, particle effects or sprites rendered at the point of impact that denote that an attack has landed (or been blocked)
- ◆ It's not vital to use different indicators for each hit-type
- ◆ It's customary to use different colors for the player and enemies
- ◆ Different visuals are needed for blocked hits and successful hits

Feedback

◆ Knockdown animations

- ◆ Needed to prevent stun-lock from hit animations
- ◆ Provides critical pacing to combat experience
- ◆ Characters are usually invulnerable while knocked down. Exceptions to this should be rare especially when used on the player (bosses, for example).

Cadence

- ◆ Rhythm of combat
 - ◆ It's the rate at which hits are thrown, the pauses between pummeling opponents, the 'beats per minute' of combat in a game
- ◆ Very hard to define rigorously
- ◆ Easy to pick up, difficult to explain

Animation considerations

- ◆ Responsiveness
- ◆ Readability
- ◆ Wow factor
- ◆ Potential trouble spots

Animation considerations

◆ Anticipation vs. responsiveness

- ◆ Anticipation and build up before an action is learned as a fundamental concept of animation
- ◆ Videogames often require this concept to be subdued or eliminated in favor of quick responsiveness
- ◆ As an example, the act of jumping in real life requires the person to crouch down and then spring forth before leaving the ground. In a videogame this 1–3 second delay would severely hamper responsiveness.
- ◆ Responsiveness trumps anticipation

Animation considerations



Animation considerations



Animation considerations

◆ Snapping issues vs responsiveness

- ◆ Characters often need to perform moves from unready states (e.g. fire an arrow without bow equipped or parry with a weapon not fully drawn).
- ◆ If the character performs WHAT the player wants him to do WHEN the player wants him to do it, minor snapping will usually not look obtrusive.

Animation considerations



Animation considerations

- ◆ Readability – Selling the animation
 - ◆ Silhouette
 - ◆ POV
 - ◆ Holding

Animation considerations

◆ Silhouette

- ◆ If your character were a complete shadow, could you still tell what he's doing?
- ◆ Applying a pure black texture or palette is a good way to test this.
- ◆ Good silhouette will help ensure that your character's actions are readable in different types of lighting and backdrop.

Animation considerations

◆ Silhouette

- ◆ It is helpful in order to preserve silhouette to keep a character's arms away from their body and to position them in a wide stance. This helps especially with caped characters.

Animation considerations



Animation considerations

◆ POV

- ◆ Is your character going to be viewed from a third person, side-scroller or top-down perspective?
- ◆ If not main character, what direction is the player going to be looking at this character from when they're performing this action?
- ◆ Play up to this view, it is important that actions look their best at the angle they're going to be seen in game. Exaggerations that might look ridiculous from other views can help increase readability from the main POV.

Animation considerations

◆ Holding

- ◆ Responsiveness dictates when animations need to be speedy. Readability dictates when they need to slow down or hold.
- ◆ Rumor has it that during filming, Bruce Lee was asked to slow down his movement as at he was practically a blur on celluloid.
- ◆ In a similar way it is often useful to hold moves at the moment of impact or in their most outstretched frame to improve readability

Animation considerations

◆ Wow factor

- ◆ Finishing moves offer a good opportunity to add flourish as they usually have longer recovery times since the opponent is knocked down.
- ◆ We want the player to enjoy a move enough to want to do it again
- ◆ No matter how cool the move is, the more we see it the less interesting it seems

Animation considerations

◆ Potential trouble spots

- ◆ Facing – Make sure that your character always returns to the same combat idle. In realistic combat your stance changes, this might or might not be inside the scope of your animation.
- ◆ Advancing animations – If attack animations translate characters forward, characters need to move back in their hit animations. Otherwise characters look like they're being pushed and not pummeled.
- ◆ Spinning moves – These require special attention in order to finish with the right facing and for linking.

Conclusion

- ◆ **Interdependencies**

- ◆ In combat systems animation is closely linked to design more than in most other areas of a game. Animation needs to account for design but it also feeds back into design.

- ◆ **Responsiveness > everything**

- ◆ When in doubt, choose to favor responsiveness

Q&A

Questions, comments?

For additional information email us at:

lbarriga@turbinegames.com

gallerani@vvisions.com